XcmsAllocColor, XcmsAllocNamedColor – allocate colors

Status XcmsAllocColor(*display***,** *colormap***,** *color_in_out***,** *result_format***)**
Display ***display;*
Colormap *colormap**;*
XcmsColor ***color_in_out;*
XcmsColorFormat *result_format**;*

Status XcmsAllocNamedColor(*display*, *colormap*, *color_string*, *color_screen_return*, *color_exact_return*, *result_format*)
Display **display*;
Colormap *colormap*;
char **color_string*;
XcmsColor **color_screen_return*;
XcmsColor **color_exact_return*;
XcmsColorFormat *result_format*;

display Specifies the connection to the X server.

colormap Specifies the colormap.

color_exact_return Returns the color specification parsed from the color string or parsed from the corresponding string found in a color-name database.

color_in_out Specifies the color to allocate and returns the pixel and color that is actually used in the colormap.

color_screen_return Returns the pixel value of the color cell and color specification that actually is stored for that cell.

color_string Specifies the color string whose color definition structure is to be returned.

result_format Specifies the color format for the returned color specification.

The XcmsAllocColor function is similar to **XAllocColor** except the color can be specified in any format. The **XcmsAllocColor** function ultimately calls **XAllocColor** to allocate a read-only color cell (colormap entry) with the specified color. **XcmsAllocColor** first converts the color specified to an RGB value and then passes this to **XAllocColor**. **XcmsAllocColor** returns the pixel value of the color cell and the color specification actually allocated. This returned color specification is the result of converting the RGB value returned by **XAllocColor** into the format specified with the result_format argument. If there is no interest in a returned color specification, unnecessary computation can be bypassed if result_format is set to **XcmsRGBFormat**. The corresponding colormap cell is read-only. If this routine returns **XcmsFailure**, the color_in_out color specification is left unchanged.

XcmsAllocColor can generate a **BadColor** errors.

The **XcmsAllocNamedColor** function is similar to **XAllocNamedColor** except that the color returned can be in any format specified. This function ultimately calls **XAllocColor** to allocate a read-only color cell with the color specified by a color string. The color string is parsed into an **XcmsColor** structure (see **XcmsLookupColor**), converted to an RGB value, and finally passed to **XAllocColor**. If the color name is not in the Host Portable Character Encoding, the result is implementation-dependent. Use of uppercase or lowercase does not matter.

This function returns both the color specification as a result of parsing (exact specification) and the actual color specification stored (screen specification). This screen specification is the result of converting the RGB value returned by **XAllocColor** into the format specified in result_format. If there is no interest in a returned color specification, unnecessary computation can be bypassed if result_format is set to **XcmsRGBFormat**. If color_screen_return and color_exact_return point to the same structure, the pixel field will be set correctly, but the color values are undefined.

XcmsAllocNamedColor can generate a **BadColor** errors.

BadColor A value for a Colormap argument does not name a defined Colormap.

XcmsQueryColor(3X11), XcmsStoreColor(3X11)
Xlib – C Language X Interface